

REMARKS

Claims 1-3, 5-9, 11-13, 15-19, 21-23 and 25-29 were rejected under 35 USC 103 based on Thornton (“Using non-Java Code”) in view of Chan US Patent 6,769,123). Claims 4, 10, 14, 20, 24 and 30 were rejected under 35 USC 103 based on Thornton, in view of Chan in view of Igra (US Patent 6,701,485). Applicants have canceled claims 1-30, and entered new claims 31-54 above. Applicants respectfully traverse the foregoing rejections as applied to the new claims 31-54, as follows.

According to the present invention, **at run time, in response to respective requests in Java format from an application to create respective proxy objects for corresponding COM objects**, a run time program module generates the respective proxy objects. Thus, the proxy objects are not created at compile time, and they are not created at run time without the respective requests from the application. Rather, they are created at run time upon request by the application for specific proxy objects. This minimizes the total amount of proxy object code to be generated and stored because proxy objects which are not needed by the application are not created.

The Examiner acknowledges that Thornton does not teach this, “Thornton does not explicitly teach the step of employing a dynamic proxy object wherein the dynamic proxy object implements an interface at run time, the interface corresponding to the method identified by the identifier.” However, the Examiner cites Chan to purportedly fill this gap, “Chan teaches employing a dynamic proxy object, (proxy object invoked at run time, Col. 3 lines 5-32 and col. 5 lines 5-10) to facilitate a call to the object, wherein the dynamic proxy object implements an interface at runtime (proxy object implements an interface that is neutral with respect to client-server mode and the standalone mode), the interface corresponding to the method identified by the identifier.” Applicants disagree with the purported characterization of Chan. Chan disclosed functional proxy objects and service objects that are generated without request by an application, presumably at compile time. “A client logic element has one or more proxy objects, each of which implement an interface. A server logic element has one or more service

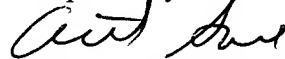
objects. Each of the service objects implements the interface.” Column 3 lines 5-10. Figure 5 illustrates multiple, apparently static proxy objects. There is no teaching or suggestion in Chan to generate proxy objects at run time **in response to respective requests from an application to create them**. Nor does Chan disclose or suggest the technique recited in claim 31 for generating the proxy objects at run time in response to respective requests from an application to create them. Therefore, claim 31 and its dependents 32-38 should be allowable.

Independent claim 39 and its dependents 40-46 distinguish over Thornton and Chan for the same reasons that independent claim 31 and its dependents 32-38 distinguish over Thornton and Chan.

Independent claim 47 and its dependents 48-54 distinguish over Thornton and Chan for the same reasons that independent claim 31 and its dependents 32-38 distinguish over Thornton and Chan.

Based on the foregoing, Applicants request allowance of the present patent application as amended above.

Respectfully submitted,



Arthur J. Samodovitz

Reg. No: 31,297